REMARKS

Claims 1-87 were pending. Claims 1, 15, 29, 67, 75, and 76 have been amended. Therefore claims 1-87 remain pending in the application subsequent entry of the present amendment.

35 U.S.C. § 112 Rejections

In the present Office Action, claims 15-28, 67, and 76-87 are rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Claims 15, 67, and 76 have been amended in a manner believed to overcome the rejections. In particular, claims 15 and 76 have been amended to point they are claiming a method. Claim 67 has been amended to point out it depends on the digital storage medium of claim 56 and not a mechanism of claim 56.

35 U.S.C. § 101 Rejections

In the present Office Action, claims 1, 3-14, 29-41, 69-71 and 75 are rejected under 35 U.S.C. § 101 as the claimed invention is directed to non-statutory subject matter. Claims 1, 29, and 75 have been amended in a manner believed to overcome the rejections.

35 U.S.C. § 102 and § 103 Rejections

In the present Office Action, claims 1-6, 8-20, 22-34, 36-47, 49-61, 63-81, and 83-87 stand rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 5,949, 994 (hereinafter "Dupree"). Claims 7, 21, 35, 48, 62 and 82 are rejected under U.S.C. § 103(a) as being unpatentable over Dupree. However, Applicant submits each of the pending claims recite features not disclosed by the cited art. Accordingly, Applicant traverses the above rejections and requests reconsideration.

For example, claim 15 recites a method for rescheduling execution or deallocating itself by a thread. As recited in claim 15, and disclosed in the Description (e.g., see Abstract), a program thread in the presently claimed invention may have an instruction within its instruction stream that accesses a parameter. When the parameter has a first value, the program thread reschedules itself in accordance with one or more conditions encoded within the parameter. Dupree teaches away from such features by disclosing a processing context that cannot reschedule itself. For example, Dupree discloses:

> "Note that the timed context may schedule or deschedule other contexts, since it has access to the scheduler bits of the 23 other processing contexts (timed context is perpetually scheduled, thus has no need for its own scheduler bit). Each of the 23 other processing contexts, however, may not schedule itself, but only deschedule itself, since these each of these contexts must already be scheduled and active to update its scheduler bit." (Dupree, col. 27 line 67 – col. 8 line 7) (emphasis added)

In Dupree, a context can only be rescheduled by a (different) timed context and not by itself. Accordingly, Dupree does not dislcose a thread rescheduling itself. In contrast, Dupree teaches threads may not reschedule themselves. For at least these reason, claim 15 is patently distinct from the cited art. The remaining independent claims are similarly distinguishable from the cited art.

In addition, the method of claim 15 recites:

> "issuing an instruction that accesses a portion of a record in a data storage device encoding one or more parameters associated with one or more conditions under which the thread is or is not to be rescheduled" (emphasis added).

Claim 15 recites the accessing of a portion of a record in a data storage device is performed by an issued instruction. In contrast, Dupree teaches control logic accesses both a scheduler latch and a timer circuit in order to determine if one of the other 23 contexts is finished with its task. In Dupree, an issued instruction does not perform the accessing. The timed context in Dupree performs the accessing when it needs to determine when to perform descheduling of one of the other 23 contexts. For descheduling, the other context must be unfinished with its task after a predetermined time limit. The timed context checks a time out condition. To perform this check, the timed context monitors via a timer circuit the predetermined time period for the active context that is performing a task, and upon reaching the time limit, the timed context generates a time-out signal. The time-out signal is generated by control logic and more specifically is the output signal of a comparator 1202 in control logic shown in FIG. 4A. Next, the timed context accesses its own context purpose (CP) register, which contains a scheduler latch for each of the other 23 contexts. Nowhere in Dupree does it disclose the timed context issues an instruction to perform the accessing of its own CP register. Dupree discloses the above steps in the following:

> "a scheduler latch is provided for each of said processing contexts except for said timed context and a context purpose register is provided for said timed context, with each of said scheduler latches being mapped into said context purpose register for said timed context in order to schedule one or more of said processing contexts for service" (Dupree, claim 1) (emphasis added)
>
> "Referring now specifically to FIG. 4A, a detailed block diagram of a timer circuit 1200 for the timed context is shown. This 24-bit timer is presettable and should be synchronized to whatever synchronizing source of time is used for the HI/O microprocessor 1012 … its value is compared against a timeout value, which is also set by the timed context (or any other suitable context) … The timer circuit 1200 includes a comparator 1202 for comparing an incrementing timer value with a preassigned timeout value. Once the timer exceeds this timeout value and timeout was selected as an interrupt source by the timed context, it will begin bidding and will immediately win priority encoding (because the timed context has the highest priority of all 24 contexts) …

> Once active, the timed context goes through a list of items to do for that moment … Items on the list could include the configuring of the sibling ports, scheduling of the other processing contexts, descheduling of any processing contexts (if they have "timed out") the generation of an interrupt to the H2 microprocessor 1014, and the running of specific subroutines … processing contexts are scheduled by setting the "scheduler" latch bit of the corresponding context. This latch is mapped into the timed context's CP register (that is the context purpose register). When the timeout period defined for that context is reached, the timed context will diagnose whether a particular context finished its task, and if it did not finish, then the timed context will "deschedule" it by resetting the same scheduler latch …
>
> Referring to FIG. 4B, a detailed block diagram of a scheduler latch mechanism is shown. In this regard, the CP register 1204 includes a separate scheduler latch 1206 for each of the processing contexts of the HI/O microprocessor 1012." (Dupree, col. 30 lines 20-66) (emphasis added)

Dupree teaches the timed context uses control logic to access "a portion of a record in a data storage device" and does not teach issuing an instruction in order to perform the access. The CP register is provided for the timed context, and therefore, the timed context would not need an instruction or interface protocol with an outside source in order to read the appropriate scheduler latch. For these further reasons, Dupree does not teach at least these features of claim 15.

Furthermore, claim 15 recites "following the conditions for rescheduling according to the one or more parameters in the portion of the record." In the present Office Action it is suggested the recited features are disclosed in col. 26, lines 3-30, and col. 27, lines 45-67 of Dupree. However, Applicant has reviewed the cited disclosures, and the remainder of the document, and finds nowhere in Dupree that an instruction accesses a portion of a record and follows the conditions for rescheduling according to the one or more parameters in the portion of the record. Both citations disclose reading a scheduler latch, but the scheduler latch is read by control logic as described above in order to determine a time-out condition and the scheduler latch does not contain one or more conditions for rescheduling or deallocation. For at least these reasons claim 15 and the remaining independent claims are patently distinguishable from the cited reference.

35 U.S.C. § 103 Rejections

Finally, as noted above, claims 7, 21, 35, 48, 62 and 82 stand rejected under U.S.C. § 103(a) as being unpatentable over Dupree. However, for at least the reasons given above, Applicant submits these claims include features neither disclosed nor suggested by the cited art.

Applicant submits all pending claims are patentably distinct from the cited art and withdrawal of the rejections is respectfully requested.

CONCLUSION

Applicant submits the application is in condition for allowance, and an early notice to that effect is requested.

Respectfully submitted,

/James W. Huffman/

Reg. No. 35,549
ATTORNEY FOR APPLICANT(S)

Date: 11/19/07